USING SAS AT NIH - ON THE MAINFRAME
SAS DISPLAY MANAGER SYSTEM

SAS programs can be run interactively either in 'line mode' or by using the full-screen SAS user interface called the SAS Display Manager System(DMS), or, informally, the Display Manager. Nowadays with the almost universal deployment of full-screen display devices and the advent of the Display Manager, line mode interactive SAS is obsolescent.

Office of Computing Resources and Services/SSS
Center for Information Technology
National Institutes of Health
Bethesda, Maryland 20892

July 1998

Contents

CIT Services and Resources

CIT'S central point of contact is the **Technical Assistance and Support Center (TASC)** of the Customer Services Branch (CSB), which provides technical and informational support to the NIH computing community and to CIT'S customers in other Federal agencies. The TASC help desk is staffed with full-time computer specialists. Users may also submit electronic mail and voice mail messages 24 hours a day and TASC will respond during regular operating hours.

Call:	4-DCRT (301-594-3278)
E-mail:	4DCRT@nih.gov
Fax:	301-402-7349
TDD:	301-496-8294
Visit:	Building 12A, Room 1011 Monday through Friday 7:30 A.M. to 5:00 P.M.
Mail:	12 South Drive MSC 5605 Bethesda, MD 20892-5605

World Wide Web access to CIT through the URL: http://www.cit.nih.gov/

TASC provides customer support for:

(1) MVS Systems

- Please refer to Interface 206, June 15, 1998, "The Evolution of WYLBUR" and "Planning is Underway for an MVS Standard System"
- MVS/ESA operating systems for both the MVS North and South systems
- **North System** (formerly ITS at the Parklawn Building)
  The North MVS System software includes JES2, TSO, ISPF, ACS, ACS WYLBUR, Model204, and relational databases.
- **South System** (formerly the MVS component of the NIH Computer Center)

This MVS component of the Enterprise Systems at the Computer Center is based on the IBM MVS/ESA (Multiple Virtual Storage/Enterprise System Architecture) Operating System with JES2 NJE Multi-Access Spool.

The MVS System customer service areas include:

Processing Tasks

- access to batch processing
  as well as interactive systems(e.g., WYLBUR, ISPF, CICS, TSO)
- language compilers
- custom printing
- automatic data backup for desktop computers
- data security protection
- disaster recovery planning
- remote job entry

Connectivity Products for Access to the MVS System

- Terminal emulation and full connectivity software for PC and Macintosh clients for telnet and dialup connections.
- Supported software packages include MS-Kermit, PROCOMM PLUS, VersaTerm, PC/TCP and OnNet, and the Protocol Conversion Facility.

Networks

- NIHnet, a high-speed network backbone that interconnects NIH Ethernet LANs, the Computer Center mainframes (MVS and Helix Systems), and the Internet.
- LAN protocols supported for NIHnet connectivity are TCP/IP, AppleTalk, IPX, XNS, and DECnet.
- Dialup access to NIHnet is available through PARACHUTE
- Internet - and international collection of networks supported by major research institutions, which communicate with each other using TCP/IP protocols. The Internet offers file transfer, remote login(telnet) and mail services.

SILK Web Facilities

- The SILK (Secure Internet LinKed) Web technologies allow users to create and run their own Web servers using NIH Computer Center hardware and software.

Relational Database Systems

- DB2
- Oracle - Oracle RDBMS on the MVS and the Oracle server Digital AlphaServer 8400

IMS

- An information management system for interactive interrogation and maintenance of large centralized data bases(restricted access).

System Tape Library via MVS on the IBM

- The Tape Library, located in Building 12, Room 1100, maintains a large supply of magnetic tapes for use with MVS.
  The Tape Library may be contacted at (301) 496-6021.

Digital AlphaServer8400

- Another component of the Enterprise Systems provides the base for a new environment for client/server(i.e. Unix-based) applications.

More detail information on the Enterprise Systems is available on the World Wide Web at http://cfb.dcrt.nih.gov/enterprise.html
To solve any problem you encounter when using the MVS System you may submit a PTR (Problem Tracking and Report System) through one of the following:

- WYLBUR **ENTER PTR** command
- World Wide Web access at http://cfb.dcrt.nih.gov/ptr.html
- E-mail to ptr@cu.nih.gov

(2) Advanced Laboratory Workstation (ALW) System

(3) Helix Systems

- Helix - a network communication system
- Convex - a computation server which runs scientific applications

In addition to the standard UNIX tools for software development, text formatting, and network communications, software packages include:

- Scientific Applications: GCG Sequence Analysis Package, Quest, BLAST, CHARMm, Gaussian, Mathematica, Prophet
- Biological Databases: GenBank, PIR, GCG, PDB
- Subroutine Libraries: IMSL - Mathematical and statistical routines
- Programming Languages: C and FORTRAN, Pascal, LISP, and C++

(4) Computer Training

The CIT Computer Training Program offers a wide variety of courses that enable users to make efficient and effective use of computing, networking, and information systems in their work. The training program is open to NIH employees and to all users of the CIT computing facilities. Course information and registration is available through CIT Home Page of the World Wide Web at **http://livrewire.nih.gov/training/looup.asp**

(5) Scientific Computing Resource Center (SCRC)

The Scientific Computing Resource Center is a shared-use computing facility for the hands-on evaluation and use of scientific software by NIH researchers.

(6) Customer Accounts

The Customer Account Office provides account and registration services to the NIH community and to customers in many other Federal agencies.

(7) Publications, Documentation and Software Distribution

CIT distributes publications and software to our customers. Documentation can be ordered by contacting TASC or online services through WYLBUR ENTER PUBWARE command or the Helix PUBWARE command.

(8) Statistical Support Staff provides technical support to NIH employees and all users of the Center for Information Technology(CIT) computing facilities.
Web site: **http://statsoft.nih.gov/**

The Statistical Support Staff provides the following services: statistical advice, assistance on the interpretation of results, technical help on the use of the supported software, documentation for the supported software, training through the CIT Computer Training Program, access to the SAS Software for Windows, Mac and OS/2 through an NIH site license

You can contact the Statistical Support Staff at 301-594-3278(4-DCRT) or send e-mail to 4dcrt@nih.gov.
Help is available Monday through Friday from 8:30 a.m. to 5:00 p.m.

Software Support on Various Computing Platforms

1) MVS/ESA Operating System

SAS/TUTOR Online Training Library, Base SAS, SAS/STAT, SAS/GRAPH, SAS/ETS, SAS/OR, SAS/FSP, SAS/AF, SAS/IML, SAS/ASSIST, SAS/CONNECT, SAS/INSIGHT, SAS/CALC, SAS/QC, SAS/TOOLKIT, SAS/ACCESS-DB2, SAS/ACCESS-ORACLE, SAS/SHARE

BMDP
A data analysis program library. Its programs perform descriptive statistics, tests for means, linear and nonlinear regression, log-linear models, maximum likelihood estimation, multivariate analysis, cluster analysis, survival analysis, and time series analysis.

SPSS
Provides capabilities for extensive file handling and data management tasks. Includes advanced statistical analysis procedures such as discriminant analysis, nonlinear and logistic regression analysis, and multivariate analysis of variance.

IMSL
This product is an extensive collection of FORTRAN-callable subroutines from Visual Numerics Inc. These routines perform mathematical and statistical computations.

GLIM
A system from the Numerical Algorithms Group Inc. for analysis of linear statistical models that can run in batch or interactive mode.

MSTAT1
This is a small collection of programs and subroutines for mathematical and statistical analysis. These programs were developed by CIT to run in batch mode.

LISREL
A program used to estimate the unknown coefficients of a set of linear structural equations.

2) SAS System Software for Windows, Macintosh, and OS/2 is available to National Institutes of Health personnel through a site license between NIH/CIT and SAS Institute Inc.

NOTE:
Request for PC, Mac, or OS/2 SAS System Software must be made to the Technical Information Office. Please call 301-594-3278(4-DCRT).

SAS products available on Windows, Macintosh, or OS/2 through this site license are:

SAS/TUTOR Online Training Library (only CBT101), Base SAS, SAS/STAT, SAS/GRAPH, SAS/FSP, SAS/AF, SAS/IML, SAS/ASSIST, SAS/CONNECT, SAS/INSIGHT, SAS/CALC, SAS/EIS, SAS/LAB, SAS/ACCESS Interface to PC File Formats, SAS/ACCESS Interface to ODBC Software, SAS/QC, SAS/OR, SAS/ETS.

New to SAS System for the Windows, Mac, and OS/2 environments is the support for ODBC. ODBC is software architecture developed by Microsoft that provides a common Application Programming Interface (API). ODBC is a Microsoft standard to allow software packages that are compliant with the ODBC specifications to exchange information. Through the use of the ODBC engine, compliant software packages such as Lotus 1-2-3, Excel, Paradox, and others can now extract information directly from SAS data sets.

SAS/ACCESS Interface to PC File Formats provides an interface to PC software products. SAS System can be used to analyze and present data directly from popular PC file formats like dBase, DIF, LOTUS 1-2-3, and Microsoft Excel.

SAS/ACCESS Interface to ODBC provides all the power and flexibility of the SAS System that one can use to analyze and present data directly from mainframe platforms, UNIX and popular PC file formats.

3) ALW System
Base SAS, SAS/STAT, SAS/GRAPH, SAS/FSP, SAS/IML, SAS/CONNECT and SAS/INSIGHT

4) Helix System(Convex server) and PC
IMSL - Mathematical and statistical routines

5) SUDAAN
SUDAAN 7.50 for Windows can be installed as a 'stand-alone' or as a 'SAS-callable'. SUDAAN is a single program consisting of a family of procedures used to analyze data from complex surveys and other observational and experimental studies involving cluster-correlated data. A complex sample may be multistage, stratified, or clustered. Many samples also have unequal probabilities of selection, or are drawn from finite populations. SUDAAN enables you to use survey data to obtain consistent estimates of population parameters and their standard errors in accordance with the sample designs. SUDAAN also produces consistent estimates of regression coefficients, descriptive statistics, and their associated standard errors for cluster-correlated and repeated measure data applications in clinical, epidemiological, toxicological, and behavioral research.

Description of SAS Products

To provide the most effective response to your SAS questions and problems, the Statistical Support Staff offers support for the following SAS products.

SAS/TUTOR Online Training Library:
Learn the SAS System as you use the SAS System with SAS Institute's library of online training courses. The SAS/TUTOR library on the mainframe currently includes six highly interactive courses. For more information, please see the CIT Computer Training brochure or call 301-594-3278.

BASE SAS:
This is the core of the SAS System. It is required for all other SAS System products. It contains the DATA step and the fundamental procedures that are needed for working with SAS data sets.

SAS/STAT:
Provides procedures for regression analysis, analysis of variance, categorical data analysis, multivariate analysis, discriminant analysis, scoring procedures, and survival analysis.

SAS/GRAPH:
Produces high-resolution graphics including several kinds of plots and charts, geographic maps, and a number of three-dimensional graphs.

SAS/ETS:
Provides procedures for time series analysis, financial report writing, linear and nonlinear systems modeling, and forecasting.

SAS/OR:
Provides a set of procedures for operations research and project management. Included in its capabilities are linear programming, Gantt charts, activity networks, and decision analysis.

SAS/FSP:
A component of the SAS System that provides interactive facilities for data entry, editing, and retrieval, printing form letters and reports, browsing external files, and building data entry and editing applications. SAS/FSP includes several procedures and a programming language (SAS Screen Control Language) for building applications. Screen Control Language (SCL) is used to write instructions that control how an application responds to user actions.

SAS/AF:
This is an interactive applications development facility that enables programmers to create interactive windowing applications. It consists of a developmental environment that includes, PROC BUILD, a programming language (SAS Screen Control Language), and an interactive full-screen debugger. Screen Control Language (SCL) is used to write instructions that control how an application responds to user actions. Object-Oriented FRAME entries, a new component of SAS/AF software enables the developers to build SAS/AF applications as graphical user interfaces. A set of predefined graphical and text objects can be used to design or paint each application window.

SAS/ASSIST:A windowing facility that lets the user run the SAS System by making selections from a series of menus. Enables users to utilize SAS with very little SAS training.

SAS/CONNECT:
Enables communications between SAS Systems running on multiple remote hosts.

SAS/INSIGHT:
Explore data analysis through a variety of interactive graphic displays.

SAS/CALC:
A interactive windowing environment that enables you to create electronic spreadsheets. You use SAS/CALC software to enter data into a spreadsheet, define formulas to compute values, generate reports and graphics, create SAS data sets, and fetch data from SAS data sets.

SAS/QC:
Provides specialized tools for quality control and design of experiments.

SAS/TOOLKIT:
Used to write SAS procedures, functions, CALL routines, formats, and informats.

SAS/SHARE:
This product allows concurrent access to SAS data sets for reading and updating. With SAS/SHARE, multiple users can gain simultaneous update access to SAS files.

SAS/IML:
An interactive matrix facility with an extensive set of mathematical and matrix operators.

SAS/ACCESS - DB2:
The SAS/ACCESS interface to DB2 can be used to analyze and present data directly from DB2.

SAS/ACCESS - Oracle:
The SAS/ACCESS interface to ORACLE can be used to analyze and present data directly from ORACLE.

A Glossary of Terms and Acronyms Used in this Handout

ddname a name(single word of user's choice), used as file reference for the operating system. The ddname can contain one to eight characters; the first character must be a letter or a national character.

libref the name used to identify a SAS data library to the SAS System. You assign a *libref* with a LIBNAME statement. The *libref* is the first-level name of a two-level name. For example, PERM is the *libref* in the two-level name PERM.CLASS.

fileref the name used to identify an external file to the SAS System. You assign a *filref* with a FILENAME statement. The *fileref* is how the SAS System references the file.

The fileref or the libref must be a valid SAS name to 'link to' or to 'point to' the file.

dsname operating system data set name
len maximum line length (used in the LRECL=*parameter*)
b block size (used in the BLKSIZE= *parameter*)
s1 primary space allocation (used in the SPACE= *parameter*)
s2 secondary space allocation (used in the SPACE= *parameter*)
iii registered initials
aaaa CIT user account
lastname user's last name

NIH Computer Center's System Managed Storage(SMS) Environment

The SMS environment operates within an all-cataloged public disk storage environment. The direct access storage is on RAMAC(3390) online disks. The various types of storage have different restrictions on the length of storage time and backup service. The FILE and TMP management classes provide permanent and short-term storage for most data sets. The MSS management class provides slightly less expensive storage for data sets that do not require backup.

NIH Computer Center's WYLBUR Data Sets

WYLBUR data sets are stored in one of two formats on the disk and the MSS.

EDIT format or LRECL format:

EDIT FORMAT
The EDIT format is the standard format used by WYLBUR to store data sets. EDIT format WYLBUR data sets are compressed to save space, while LRECL format data sets are not. EDIT format is the default storage format. If a WYLBUR data set is saved using the WYLBUR **SAVE** command, it is stored in EDIT format.

```
? SAVE AS dsname
```

EXAMPLE:

```
? COLLECT CLEAR
    1.   ? 4 5
    2.   ? 0 1
    3.   ? 2 6
? ***
? SAVE AS RAWDATA1
'RAWDATA1' SAVED AND CATALOGED ON FILE
```

To determine if a WYLBUR data set on the disk is stored in EDIT format, use WYLBUR's **SHOW DSNAME** command.

```
? SHOW DSNAME dsname
```

EXAMPLE:

```
? SHOW DSNAME RAWDATA1
FILE

   RAWDATA1 ON DSA106

   UNIT=FILE, DEVTYPE=3010200F, NO. OF VOLUMES=1
   CREATED 06/03/97, LAST USED 06/03/97
   1 TRACK (1 USED), NO.  OF AREAS=1, SECONDARY SPACE=1 TRACK
   DSORG=PS, RECFM=U, LRECL=11476, BLKSIZE=11476
   KEYLEN=0, RKP=0, OPTCD=C(20)
   EXPIRATION DATE=00/00/00, NO PASSWORD
```

If the WYLBUR data set is stored in EDIT format, this command will report **RECFM=U, LRECL=11476, and BLKSIZE=11476**.

To determine if a WYLBUR data set on the MSS is stored in EDIT format, use WYLBUR's

? SHOW DSNAME *dsname* **on MSS**

If the WYLBUR data set is stored in EDIT format, this command will report **RECFM=U, LRECL=11476, and BLKSIZE=11476**.

LRECL Format

To save a WYLBUR data set on the disk in LRECL format, add the LRECL option to WYLBUR's **SAVE** command.

? **SAVE AS** dsname **LRECL=len**
where 'len' is greater than or equal to the length of the longest data line.

EXAMPLE:

```
? COLLECT CLEAR
    1.   ? 4 5
    2.   ? 0 1
    3.   ? 2 6
? ***
? SAVE AS RAWDATA1 LRECL=6
'RAWDATA1' SAVED AND CATALOGED ON FILE
```

To determine if a WYLBUR data set on disk is stored in LRECL format, use WYLBUR's **SHOW DSNAME** command.

EXAMPLE:

```
? SHOW DSNAME RAWDATA1
FILE

   RAWDATA1 ON DSA106

   UNIT=FILE, DEVTYPE=3010200F, NO. OF VOLUMES=1
   CREATED 06/03/97, LAST USED 06/03/97
   1 TRACK (1 USED), NO.  OF AREAS=1, SECONDARY SPACE=1 TRACK
   DSORG=PS, RECFM=FB, LRECL=6, BLKSIZE=11472
   KEYLEN=0, RKP=0, OPTCD=C(20)
   EXPIRATION DATE=00/00/00, NO PASSWORD
```

SAS Reads Raw Data with LRECL Format on the Disk and the MSS

SAS **does not** read raw data files saved in EDIT format. Therefore, if your data set is saved in EDIT format, you will need to apply the following:**1**) Use WYLBUR's **USE FROM** and **RESAVE** commands on the data set

```
? USE FROM dsname
? RESAVE LRECL=len
```

where 'len' is greater than or equal to the length of the longest data line.

Estimate LRECL, BLOCK SIZE and SPACE Requirements

To output raw data from SAS, the logical record length, block size, and space allocations must be determined.

- secondary space allocation (’s2’) can be 1 or approximately 10% of the primary space allocation(’s1’).
- LRECL equal to an integer greater than or equal to the length of the longest record.

Example: Compute BLKSIZE… given LRECL=133 and output 500 records to a raw data file.

WYLBUR’s ENTER DISKCALC Command

```
? Enter Diskcalc

DISKCALC Command Procedure:

 Select a function from the following menu (or strike ENTER):
 (For fixed length, NON-keyed data sets)
 (Function 2 is applicable to all data set types)
  1   What is the Computer Center recommendation for BLKSIZE for a given LRECL?

  2   Compute how many physical records (blocks) may be stored on a track or cylinder with a given
      BLKSIZE.

  3   Show records and blocks per track and cylinder for a selection of possible BLKSIZES for a
      given LRECL.

  4   Compute total space needed to contain a specified number of records of a given LRECL for a
      selection of possible BLKSIZES.

  5   Change default disk model (currently 3390-3, 3,339 cylinders per volume, 15 tracks/cylinder,
      56,664 bytes per track).
 END Return to WYLBUR.

     Select a function (or press ENTER to see the function menu): 4Enter the LRECL (logical
     record length) of the record (or press Enter):
     LRECL: 133

     Enter the number of logical records (or press ENTER):
     Number of records: 500
     ( press ENTER )

     Space required for 500 records with LRECL=133 on a 3390-3 volume which contains
     approximately 3,339 cylinders per volume.
                    Number of      Number of
      BLKSIZE       Tracks         Cylinders
     -----------------------------------
     3,724           2
     4,123           2
     4,522           2
     5,054           2
     5,719           2
     6,517           2
     7,448           2
     8,778           2
     10,773          2
     13,566          2  <=NIH Recommendation
     18,354          2
     27,930          2
 Select a function (or press ENTER to see the function menu): END
 DISKCALC  terminating.
```

Result: NIH’s recommendation is: **LRECL=133, BLKSIZE=13,566**.

Estimate the Size of a SAS Data Set

When storing a SAS data set, the record format (FB), the logical record length (LRECL), and block size (BLKSIZE) are set automatically.
However, you **must determine** the space allocation.
General Form of the SPACE parameter: SPACE=(TRK,(s1,s2),RLSE)

Space can be requested in terms of tracks or cylinders.
TRK specifies that the space is to be allocated in units of tracks.
CYL specifies that the space is to be allocated in units of cylinders. *(15 tracks per cylinder.)*
RLSE indicates that space not used by the WYLBUR data set is to be released; RLSE should be specified whenever possible.

You can follow these steps to obtain a rough estimate of how much space you need on the disk or the MSS to store SAS data sets.

Step 1. To determine the length of each variable in the SAS data set, create a SAS data set that contains 1 observation.

```
FILENAME  IN1  'aaaaiii.dsname'  ;
DATA TEST;
  INFILE IN1  obs=1 ;
  INPUT  variable list   ;
PROC CONTENTS;
```

Step 2. Based on the results of the PROC CONTENTS procedure, determine the total length of all variables for each obseration.
(*Length of each variable can range from 2 to 8 for numeric variables and from 1 to 200 for character variables.)*

Step 3. Apply the following formula to estimate the primary space parameter.

Let X = (total length of all variables)*(total number of observations)
(The "total length of all variables" is specified as the "observation length" in the PROC CONTENTS Output.)

Let Y1 = X/6144. Let Y2 be Y1 rounded up to the next integer.
Let Z = Y2/7. The primary space allocation 's1' is Z rounded up to the next integer.

Example:
Suppose you have 12 numeric variables (each stored in 8 bytes) and 5000 observations.

```
X  = 8*12*5000 = 480,000
Y1 = 480,000/6144 = 78.125
Y2 = 79
Z  = 79/7 = 11.29
s1 = 12 tracks
```

Result: SPACE=(TRK,(12,1),RLSE).

A 'rule of thumb'
The primary space parameter('s1') - set as the larger of the two parameters.
The secondary space parameter ('s2') - set approximately 10% of the primary space allocation(10% of the 's1' parameter).

Invoking the SAS Display Manager on the Mainframe at NIH

SAS programs can be run interactively by using the full-screen SAS user interface called the SAS Display Manager System(DMS), or, informally, the Display Manager.

The SAS Display Manager System is an interactive, windowing environment in which actions are performed through a series of commands or function keys. The Display Manager can be used to prepare and submit SAS programs, view and print the results, and debug and resubmit the programs.

Full-screen access to the MVS operating system can be accomplished through TN3270 or NIHnet 3270 Protocol Converter.

Once connectivity is made you initiate an interactive SAS session by issuing the following command at the TSO (READY) prompt

ex 'zabcrun.sas.clist(sas)'

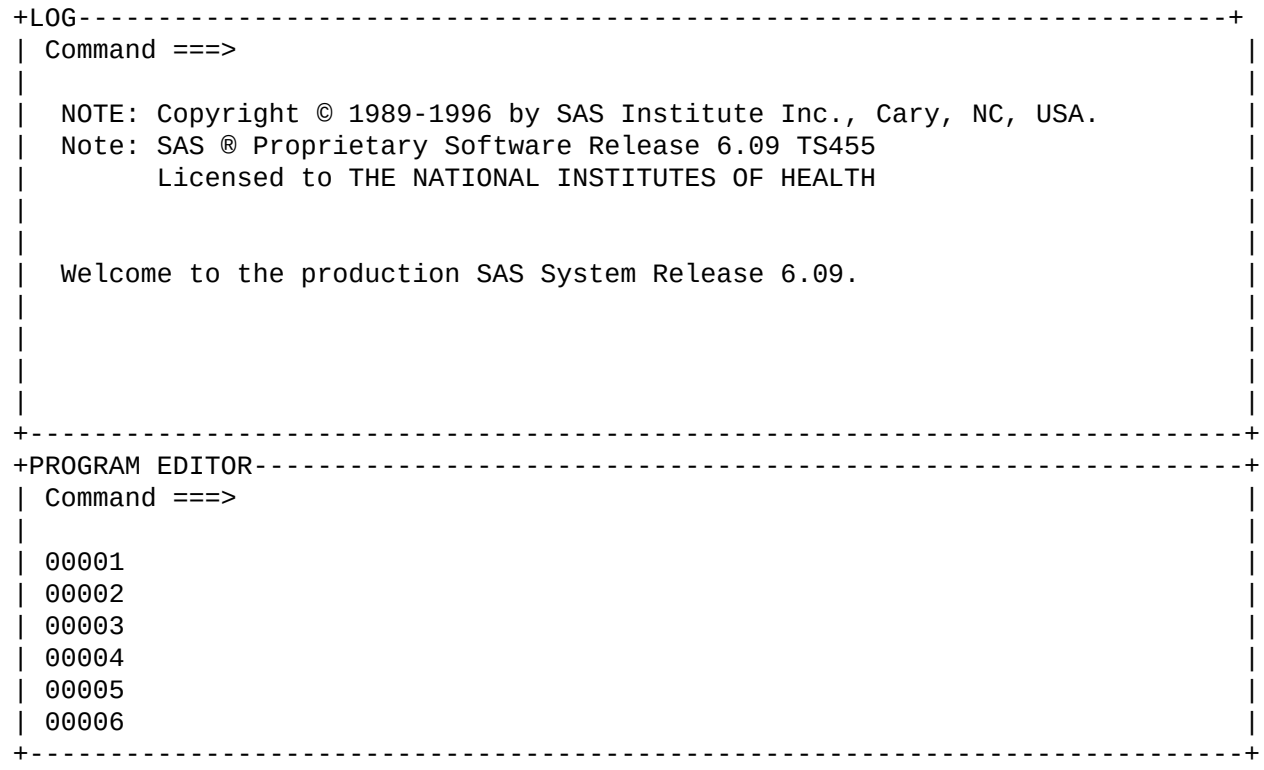
Display Manager Windows

The SAS Display Manager System is an interactive windowing environment from which you can perform many basic tasks, including those listed below.

Task	Window Name
Edit and execute program statements	PROGRAM EDITOR
Check the contents of the SAS log	LOG
Browse procedure output	OUTPUT
Set function keys	KEYS
Get help about the SAS System	HELP

Program Editor, Log and Output Windows

When you invoke the SAS System three primary windows are displayed. Theses are the LOG window, the PROGRAM EDITOR window and the OUTPUT window.



Default Display Manager Windows

The LOG window displays messages about your session and about programs you submit.
The PROGRAM EDITOR window is where you

- enter, edit, and submit program statements
- include saved SAS programs

The OUTPUT Window

- is automatically displayed if your program produces output
- enables you to browse procedure output from your current SAS session.

The default OUTPUT Window

Activate Window

Only one window can be active at a time. To activate a window you can

- position the cursor anywhere within the window and press ENTER
- use its name in a window-call command

Commands used to activate display manager windows:

Command	**Window Activated**
PGM	PROGRAM EDITOR
LOG	LOG
OUTPUT, OUT	OUTPUT
KEYS	KEYS
HELP	HELP

END closes any window except the LOG and PROGRAM EDITOR windows.

Issue Display Manager commands

You may issue display manager commands by

- entering the command on the command line

or

- using the function keys

Listed are a few function key definitions. Press the function key F11 to view other function key definitions.

You can redefine a function key by typing over the command displayed in the KEYS window.

Issue the END command to close the KEYS window

Function KEYS

KEY	DESCRIPTION	ACTION
F11	KEYS	Display the KEYS window
F12	UNDO	Undo edits
Shift F3	HELP	Display the HELP window
Shift F4	ZOOM	Enlarge the active window, or return window to its original size
Shift F5	SUBMIT	Submit SAS statements
Shift F6	RECALL	When you issue the RECALL command, programs are redisplayed starting with the most recently submitted program. The recall buffer is where submitted SAS code is stored and remains for the duration of the SAS session. **To store the SAS program in an external file, issue the FILE command.**
Shift F9	BACKWARD	Scrolls the content of a window backward(toward the top)
Shift F10	FORWARD	Scrolls the content of a window forward(toward the bottom)
Shift F11	OUTPUT	Display OUTPUT window

Issue Display Manager commands on the command line

Command	Action
PGM	Move and activate the PROGRAM EDITOR window
LOG	Move and activate the LOG window
OUTPUT or OUT	Move and activate the OUTPUT window
RECALL or REC	Recall submitted code
SUBMIT or SUB	Submit SAS program
CLEAR	Clear window
ZOOM or Z	Increase or decrease window size
INCLUDE ‘aaaaiii.filename’ where aaaa - registered CIT account iii - registered CIT initials filename - external file name	Include text (SAS program or data file) in the PROGRAM EDITOR window
FILE ‘aaaaiii.filename’ where aaaa - registered CIT account iii - registered CIT initials filename - external file name	Save PROGRAM EDITOR, LOG, or OUTPUT window
LIBNAME or LIB	Open Library window and lists active SAS libraries
BYE or ENDSAS	Exit SAS Display Manager
BACKWARD or UP	Scroll backward
FORWARD or DOWN	Scroll forward
TOP	Scroll to the top of the window
BOTTOM	Scroll to the bottom of the window
END	Close windows
HELP	Activate HELP window
KEYS	Open and activate the KEYS window
LEFT	Scroll the window to the left
RIGHT	Scroll the window to the right
?	Recall last command

Full-Screen Editing with Text Editor commands

The PROGRAM EDITOR window provides a means to edit the program with text editing commands.

Command	Action
Cn	copy n lines before or after the line designated with A or B command.
Dn	delete n lines.
In	insert n lines after the current line.
Mn	move n lines to the place designated with A or B command.
CC	copy a block of lines to the place designated with A or B command.
DD	delete a block of lines.
MM	move a block of lines to the place designated with A or B command.
A	specifies that the line(s) to be moved, copied, or inserted be placed **after** this line.
B	specifies that the line(s) to be moved, copied, or inserted be placed **before** this line.

Examples - Text Editing commands

- To insert a new line for entering text, type **I**, the insert line command, on any part of a line number.

00**I**001 line one
000002 line two
000003 line three
and press ENTER. One new line is inserted between first and second lines. Change is seen below.

000001 line one
000002
000003 line two
000004 line three

- To move or copy single lines or blocks of lines, you need to indicate the location where the text is to be moved. The following lines show how to use the **M** (move) and **A** (after) line commands:

M00001 Move this line
A00002 after the second line
000003 with the **M** (move) and **A** (after) line commands.

When you press ENTER, the first line becomes the second line. Change is seen below.

000001 after the second line
000002 Move this line
000003 with the **M** (move) and **A** (after) line commands.

- You can also use line commands to affect blocks of lines. Indicate **CC** on the line numbers of the first and last lines of a block of lines to be copied.

CC0001 Copy the first two lines
CC0002 after the third line
A00003 with the **CC** (copy) and **A** (after) line commands.

When you press ENTER, lines one and two are copied after line three. Change is seen below.

000001 Copy the first two lines
000002 after the third line
000003 with the **CC** (copy) and **A** (after) line commands.
000004 Copy the first two lines
000005 after the third line.

- The move (**M** or **MM**) commands work just like the copy (**C** or **CC**) commands except that lines are moved instead of copied.

Submit a SAS Program

Enter the SAS statements in the **PROGRAM EDITOR** window and issue the **SUBMIT** command.

When the program begins to execute, the following will occur:

- SAS statements disappear from the **PROGRAM EDITOR** window.
- A status line appears in the top right corner of the window, indicating what type of program (a DATA step or a PROC step) is running.

While your program is executing, the following information is written to the **LOG** window:

- the submitted SAS statements
- information about the data being read, including the location of the data, the number of records read, and the maximum record length
- confirmation that the program ran successfully, or error messages

Clearing Messages

Messages accumulate in the **LOG** window during the current SAS session.

You can clear the contents of a window by issuing the **CLEAR** command.

The CLEAR Command

The general form of the CLEAR command is

CLEAR PGM|LOG|OUTPUT|RECALL

```
CLEAR PGM         clears the PROGRAM EDITOR window
CLEAR LOG         clears the LOG window
CLEAR OUTPUT      clears the OUTPUT window
CLEAR RECALL      clears the recall buffer
```

If you do not specify a window name, the active window is cleared. A message lets you know that the window has been cleared.

Recall your program

The recall buffer is the area where submitted SAS code is stored for the duration of your SAS session.

Issue the **RECALL** command from the **PROGRAM EDITOR** window to recall the most recently submitted SAS program.

Once the data set is created, you **do not** need to submit the DATA step again.

Managing SAS Output

You can use the **OUTPUT MANAGER** window to manage the output accumulated in the **OUTPUT** window.

Issue the **MANAGER** command to open the **OUTPUT MANAGER** window.

After you press **ENTER**, the **OUTPUT MANAGER** window is active.

Selection field commands in the **OUTPUT MANAGER** window include

S to view selected pages.

E to edit selected pages.

P to print selected pages.

D,V to delete selected pages.

F to send selected pages to a file.

Issue the **MANAGER OFF** command to close the **OUTPUT MANAGER** window.

Save a SAS program or text file

Issue the FILE command to store a SAS program or a data file

The general form of the FILE command is

FILE 'aaaaiii.filename'

where aaaa - the CIT account
iii - the CIT initials
filename - the name of the external file

To store a copy of a program as a member of a partitioned data set use

FILE 'aaaaiii.filename(member)'

where filename - the partitioned data set name
member - the member name

Include a Saved Program

Issue the INCLUDE command to include a copy of a stored program into the PROGRAM EDITOR window.

The general form of the INCLUDE command is

INCLUDE 'aaaaiii.filename'

where aaaa - the CIT account
iii - the CIT initials
filename - the name of the external file

To include a copy of a program stored as a member of a partitioned data set use

INCLUDE 'aaaaiii.filename(member)'

where filename - the partitioned data set name
member - the member name

End the SAS Session and Logoff

To end a SAS Session...

1. Issue the **BYE** or **ENDSAS** command from any display manager window.

2. Enter **Logoff** at the **READY** prompt.

READY (enter **LOGOFF**)

Sample SAS Programs

NOTE:
All FILE, TMP, and MSS data sets are cataloged. All references to data sets on the disk are made through the catalog.

Reading from External Files

To read a data set in the SAS Display Manager System it must satisfy three requirements:

- must be on disk or on the MSS
- must be saved in LRECL format
- **cannot** exist on tape

Input

1. Read raw data when the input data immediately follows a CARDS statement in the SAS program

```
DATA TEST;
INPUT variable list ;
CARDS;
(data lines)
;
```

2. Read raw data in LRECL format on disk and on the MSS

```
FILENAME filref 'aaaaiii.dsname' ;

DATA TEST;
  INFILE fileref ;
  INPUT variable list ;
```

- **Or without the FILENAME statement**

```
DATA TEST;
  INFILE  'aaaaiii.dsname' ;
  INPUT variable list ;
```

3. Read SAS data set on the disk and on the MSS

```
LIBNAME libref 'aaaaiii.dsname';

DATA TEST;
  SET libref.SASname;
```

Output

4. Write raw data to a file on the disk

```
FILENAME fileref 'aaaaiii.dsname'
UNIT=FILE  DISP=(NEW,CATLG)
        SPACE=(TRK,(s1,s2),RLSE)  RECFM=FB  LRECL=len  BLKSIZE=b  ;

      DATA _NULL_;
         /* other-SAS-statements */
      FILE fileref;
         /* other-SAS-statements */
```

NOTE:
Specify DISP=MOD if the data set exists and new records are to be added to the end.

5. Write raw data to a file on the MSS

```
      FILENAME fileref 'aaaaiii.dsname' UNIT=MSS  DISP=(NEW,CATLG)
      SPACE=(TRK,(s1,s2),RLSE)  RECFM=FB  LRECL=len  BLKSIZE=b ;

      DATA _NULL_;
         /* other-SAS-statements */
      FILE fileref;
          /* other-SAS-statements */
```

6. Write SAS data set to the disk

```
      LIBNAME libref 'aaaaiii.dsname' UNIT=FILE
         DISP=(NEW,CATLG)  SPACE=(TRK,(s1,s2),RLSE) ;

      DATA libref.SASname;
         /* other-SAS-statements */
```

7. Write SAS data set to the MSS

```
LIBNAME libref 'aaaaiii.dsname' UNIT=MSS
DISP=(NEW,CATLG)
         SPACE=(TRK,(s1,s2),RLSE) ;

      DATA libref.SASname;
         /* other-AS-statements */
```

SAS Format Library

User-written formats and informats are stored in a type of SAS file called a *catalog.*

User-written informats and formats are either temporary or permanent, depending on whether they are stored in a temporary or permanent catalog.

Temporary Formats and Informats:

Formats and informats are temporary when you **do not** specify the **LIBRARY =** option in the PROC FORMAT statement. By default, the user-defined formats are stored in a temporary catalog named WORK.FORMATS. Consequently these formats disappear when SAS terminates. Therefore, you can use temporary informats and formats only in the same SAS job or session in which they are created.

Example 1:

```
PROC FORMAT ;     /* There is no LIBRARY = option, so formats and informats
                     would be saved in the catalog WORK.FORMATS  */
```

Permanent Formats and Informats:

Example 2:

```
LIBNAME  LIBRARY  'aaaaiii.dsname' DISP=(NEW,CATLG) SPACE=(TRK,(s1,s2))
UNIT=FILE ;

PROC FORMAT  LIBRARY=LIBRARY ; /* Formats or informats stored in a catalog
                                  named FORMATS in the SAS library LIBRARY.
                                  It is recommended, but not required, that
                                  you use the word LIBRARY as the libref */
```

Example 3:

```
LIBNAME  abc  'aaaaiii.dsname' DISP=(NEW,CATLG) SPACE=(TRK,(s1,s2))
UNIT=FILE;

PROC FORMAT  LIBRARY= abc ;    /* Formats or informats stored in a catalog
                                  named FORMATS in the SAS library abc. If
                                  there isn't a catalog named FORMATS in the
                                  library abc, the SAS System creates one.*/
```

Example 4:

```
LIBNAME  out1 'aaaaiii.dsname' DISP=(NEW,CATLG) SPACE=(TRK,(s1,s2))
UNIT=FILE;

PROC FORMAT  LIBRARY= way.cool ; /* Formats or informats stored in a
                                    catalog named cool in the library
                                    way */
```

Example 5: Create a permanent format catalog on disk

```
LIBNAME LIBRARY ’aaaaiii.dsname’ UNIT=FILE  DISP=(NEW,CATLG)
SPACE=(TRK,(s1,s2)) ;

PROC FORMAT LIBRARY=LIBRARY ;
     value   format name
       (enter specifications)
     ;
```

Example 6: Use formats from a permanent format catalog on disk

Permanent formats and informats can be used in the program that creates them and in subsequent SAS jobs or sessions.
In DATA and PROC steps following the PROC FORMAT step, the SAS System searches the **LIBRARY** library for the **FORMATS** catalog file.

If you wish to review the entries in the format library the **FMTLIB** option may be used to display the contents of a permanent format catalog

```
LIBNAME LIBRARY  ’aaaaiii.dsname’ ;

PROC FORMAT LIBRARY = LIBRARY FMTLIB ;
```

How to…use formats created in a permanent SAS format library

Method 1:

Use formats created in a permanent SAS format library by including a LIBNAME statement with a libref LIBRARY

```
libname library ’aaaiii.dsname’;
  proc freq data=library.demog;
    table race*sex;
    format race race. sex sex.;
```

Method 2:

In general, you can use any libref in the LIBNAME statement.
If you do not use the libref LIBRARY then use the SAS option **FMTSEARCH=** to specify the libref you used.

```
libname  in  ’aaaaiii.dsname’;
  options  fmtsearch = (in);
  proc freq data=in.demog;
     table race*sex;
     format race race. sex sex.;
```

Method 3:

To access more than one format library include one LIBNAME statement for each library then list the librefs after the **FMTSEARCH=** option separated by spaces.

```
  libname in1 ’aaaaiii.dsname1’;
  libname in2 ’aaaaiii.dsname2’;
  options  fmtsearch = (in1 in2);
```

SAS/TUTOR Online Training Software

Through the **SAS/TUTOR Online Training Software** you can practice new skills as often as you like while using the SAS Display Manager. These courses include lessons with questions, guided practice, quizzes, and optional reviews. Menus, a course index, and suggested learning paths enable you to scan for selected topics quickly and conveniently. You choose your own instructional path, controlling the time you spend on each topic. The courses include a SAVE button that enables you to mark your place when you exit the course to return later.

1 - Fundamentals of the SAS System

2 - Reading Raw Data and Formatting Values with the DATA Step

3 - Creating, Modifying, and Processing Variables with the DATA Step

4 - Developing Custom Data Entry Applications

5 - Creating and Enhancing SAS/GRAPH Output

6 - Creating Tables with PROC TABULATE

1- Fundamentals of the SAS System

This course is for new SAS software users who want to process data, produce reports and graphics, perform ad hoc queries, create and modify SAS data sets. This course teaches you how to get started using the SAS System to perform common data processing tasks. Approximate time to complete the course is 12 to 15 hours.

Overview of the SAS System	30-50 min.
Module 1(includes 3 Lessons)	3 hours
Module 2(includes 4 Lessons)	5 hours
Module 3(includes 3 Lessons)	3 hours
Module 4(includes 2 Lessons)	2 hours

2 - Reading Raw Data and Formatting Values with the DATA Step

This course is for beginning SAS software users who want to learn a variety of methods for reading raw data and formatting variable values. This course is also appropriate for experienced SAS software users who work with complex data files. Before selecting this course, you should be able to recognize SAS data set variables, observations, create a SAS data set using the DATA Step, and read a SAS data set using the SET statement. Lessons 1 through 5 can each be completed in about an hour. Each section in Lesson 6 takes approximately 45 minutes to complete.

3 - Creating, Modifying, and Processing Variables with the DATA Step

This course is for SAS software users who are familiar with the DATA Step but want to expand their knowledge of the syntax and structure of DATA Step programming. The course teaches the basic skills needed to create new variables and modify or process existing variables and variable values using the DATA Step. Before selecting this course, you should be familiar with the basic concepts of the SAS System. Specifically, you should be able to create a SAS data set using the DATA Step and read a permanent SAS data set using the SET statement. Each of the five lessons can be completed in about an hour.

4 - Developing Custom Data Entry Applications

This course is for SAS System users who want to develop custom data entry applications with SAS/FSP software. You learn how to build a data entry application with the FSEDIT procedure and customize an FSVIEW session. Before selecting this course, you should be familiar with base SAS software and be able to enter and edit data with the FSEDIT and FSVIEW procedures in SAS/FSP software. Each of the six lessons can be completed in about an hour.

5 - Creating and Enhancing SAS/GRAPH Output

This course is for beginning and experienced SAS software users who want to produce high-quality presentation graphics with SAS/GRAPH software. You learn how to create basic plots and charts, enhance them individually and globally, set up the graphics environment, and manage graphics output. To begin this course at an introductory level, you should be familiar with the basic concepts of the SAS System. To begin at a more advanced level, you should know how to use SAS/GRAPH software to create plots, bar charts, and pie charts. Each of the six lessons can be completed in about an hour.

6 - Creating Tables with PROC TABULATE

This course is for users who want to analyze and present data effectively in table form. You learn to use PROC TABULATE to organize data in tables, request statistics, and enhance output. To begin the course at an introductory level, you should be familiar with the basic concepts of the SAS System. To begin at a more advanced level, you should know how to create basic one-, two-, and three-dimensional tables, including how to specify class and analysis variables, arrange statistics and variables in tables, create summary columns and rows, and format summary values and table cells. Each of the five lessons can be completed in about an hour.

SAS/TUTOR Online Training Software is a full-screen application and requires full-screen access to the MVS system. The full screen access can be accomplished by means of Internet or dial-up access. Connectivity questions should be directed to 4-DCRT(301-594-3278).

Log on

Bold italicized entries represent CIT user registered id.

Enter command or HELP:

LOGON 3270
INITIALS? ***iii***
ACCOUNT? ***aaaa***

Enter RACF Password for RACF iii

PASSWORD? ***pppp***
TERMINAL? (Press ENTER)
BOX? (Press ENTER)

aaaiii LOGON IN PROGRESS AT hour:minute:second ON mmm dd,yyyy
NIH/CIT/CFB TSO

READY

Enter the following command at the cursor:

EX 'ZABCRUN.SAS.CLIST(SASTUTOR)' (Press ENTER)